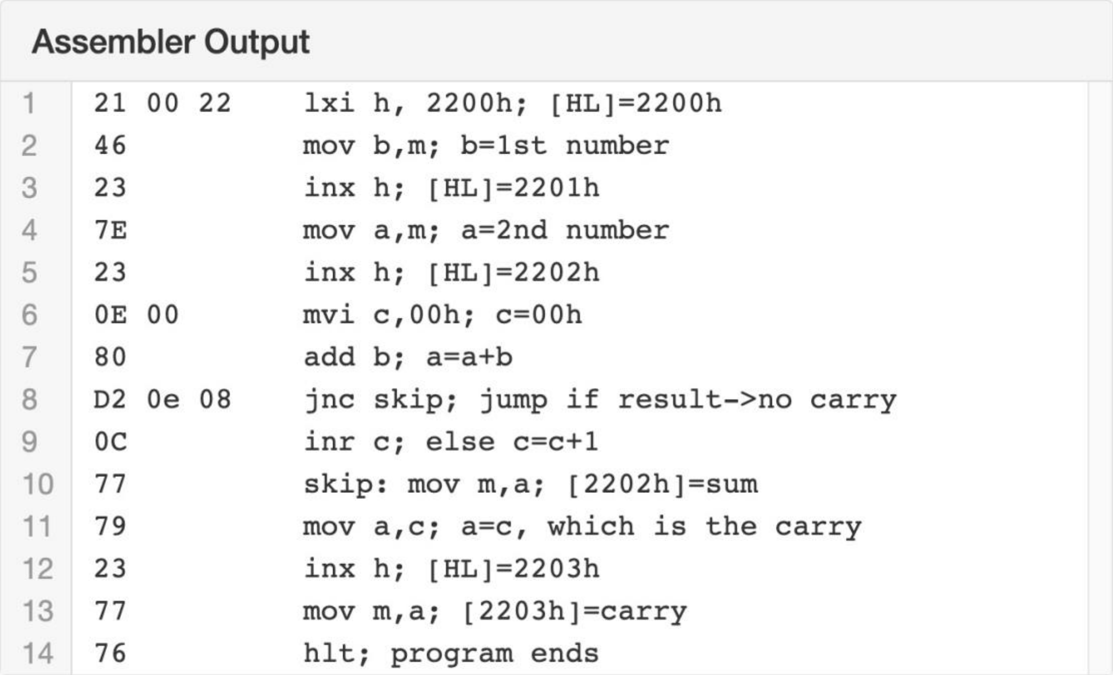
ASSIGNMENT 1

NAME: SAILIK PANDEY

ROLL: 002110501132

SEC - A3 BCSE II

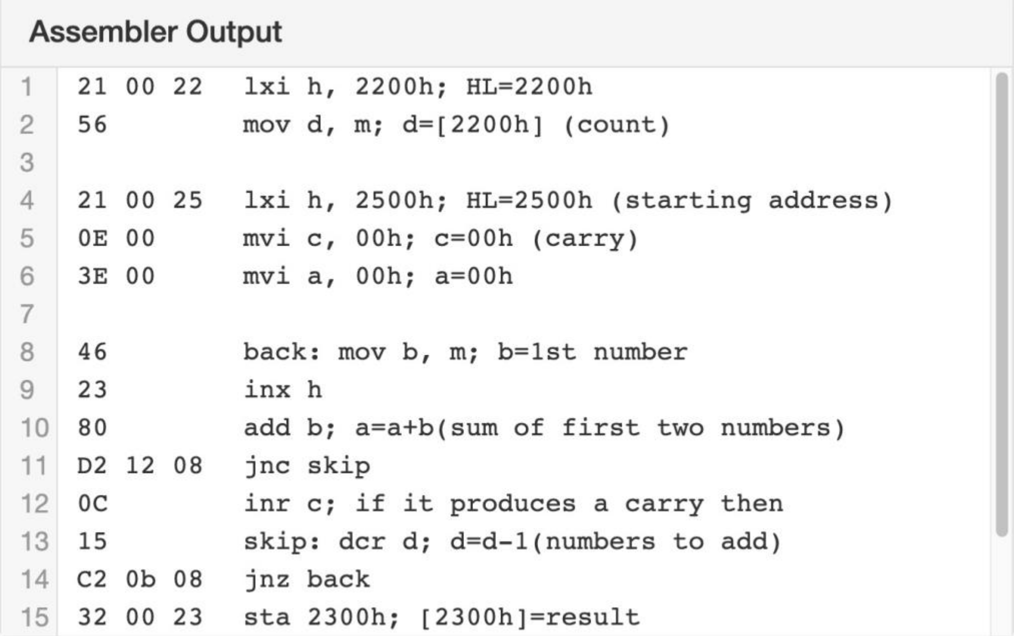
1. Load the contents of the memory locations 2200H and 2201H into registers. Add these registers and store the result in memory locations 2202H and 2203H.



Assembler Output

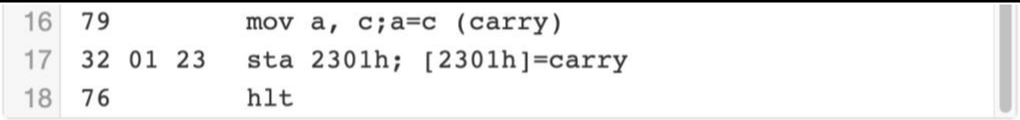
```
21 00 22    lxi h, 2200h; [HL]=2200h
46          mov b,m; b=1st number
23          inx h; [HL]=2201h
7E          mov a,m; a=2nd number
23          inx h; [HL]=2202h
0E 00       mvi c,00h; c=00h
80          add b; a=a+b
D2 0e 08    jnc skip; jump if result->no carry
0C          inr c; else c=c+1
77          skip: mov m,a; [2202h]=sum
79          mov a,c; a=c, which is the carry
23          inx h; [HL]=2203h
77          mov m,a; [2203h]=carry
76          hlt; program ends
```

2. Find the sum of N numbers stored in consecutive locations starting from 2500H. The value of N is stored in 2200H. Store the result in locations 2300H and 2301H.



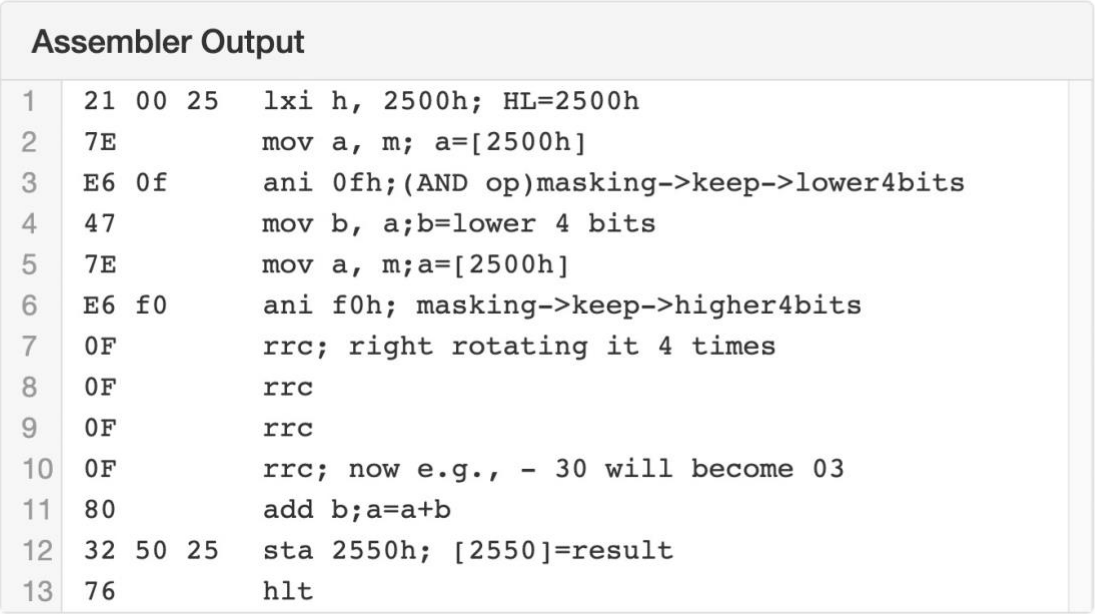
Assembler Output

```
21 00 22   lxi h, 2200h; HL=2200h
56         mov d, m; d=[2200h] (count)

21 00 25   lxi h, 2500h; HL=2500h (starting address)
0E 00      mvi c, 00h; c=00h (carry)
3E 00      mvi a, 00h; a=00h

46         back: mov b, m; b=1st number
23         inx h
80         add b; a=a+b(sum of first two numbers)
D2 12 08   jnc skip
0C         inr c; if it produces a carry then
15         skip: dcr d; d=d-1(numbers to add)
C2 0b 08   jnz back
32 00 23   sta 2300h; [2300h]=result
```



```
79         mov a, c;a=c (carry)
32 01 23   sta 2301h; [2301h]=carry
76         hlt
```

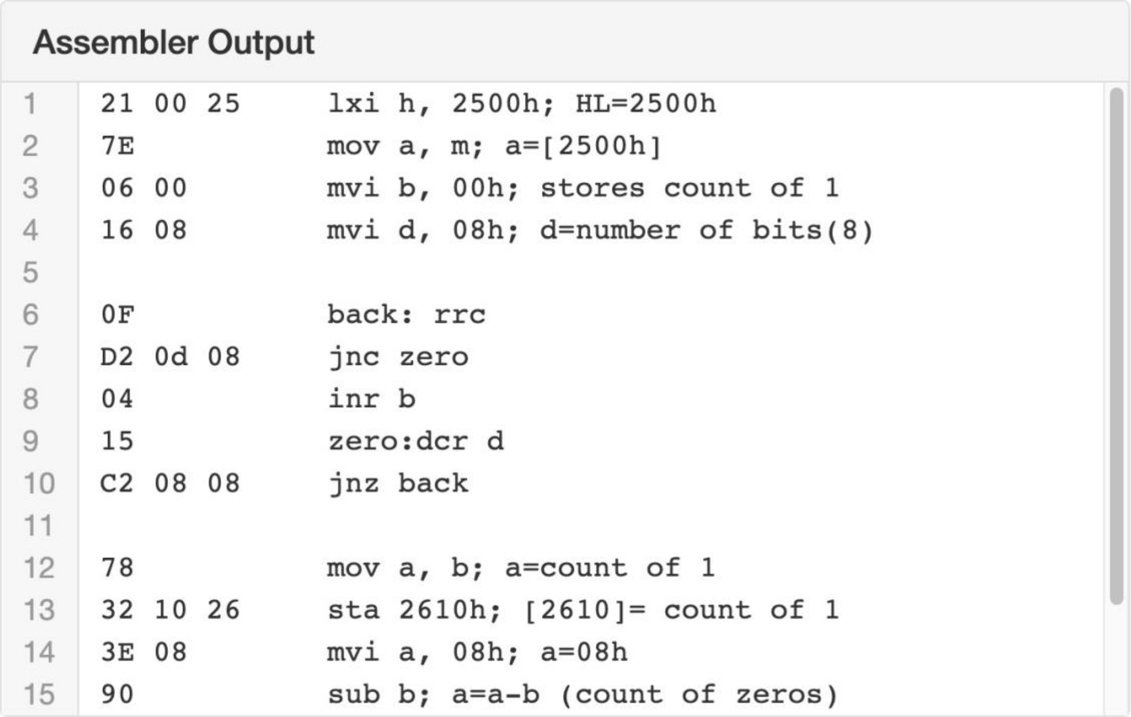
3. Find the sum of the least significant 4 bits and most significant 4 bits of a byte stored in memory location 2500H. Store the result in 2550H.



Assembler Output

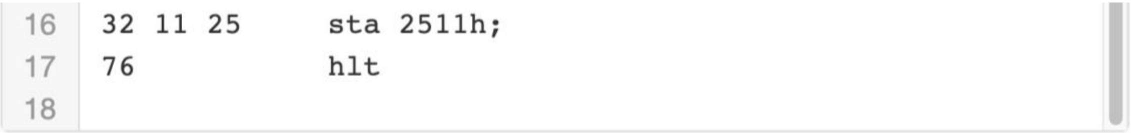
```
21 00 25   lxi h, 2500h; HL=2500h
7E         mov a, m; a=[2500h]
E6 0f      ani 0fh;(AND op)masking->keep->lower4bits
47         mov b, a;b=lower 4 bits
7E         mov a, m;a=[2500h]
E6 f0      ani f0h; masking->keep->higher4bits
0F         rrc; right rotating it 4 times
0F         rrc
0F         rrc
0F         rrc; now e.g., - 30 will become 03
80         add b;a=a+b
32 50 25   sta 2550h; [2550]=result
76         hlt
```

4. Write a program to count the ‘1’s and ‘0’s of a byte stored in 2500H. Store the result in 2610H and 2511H, respectively.



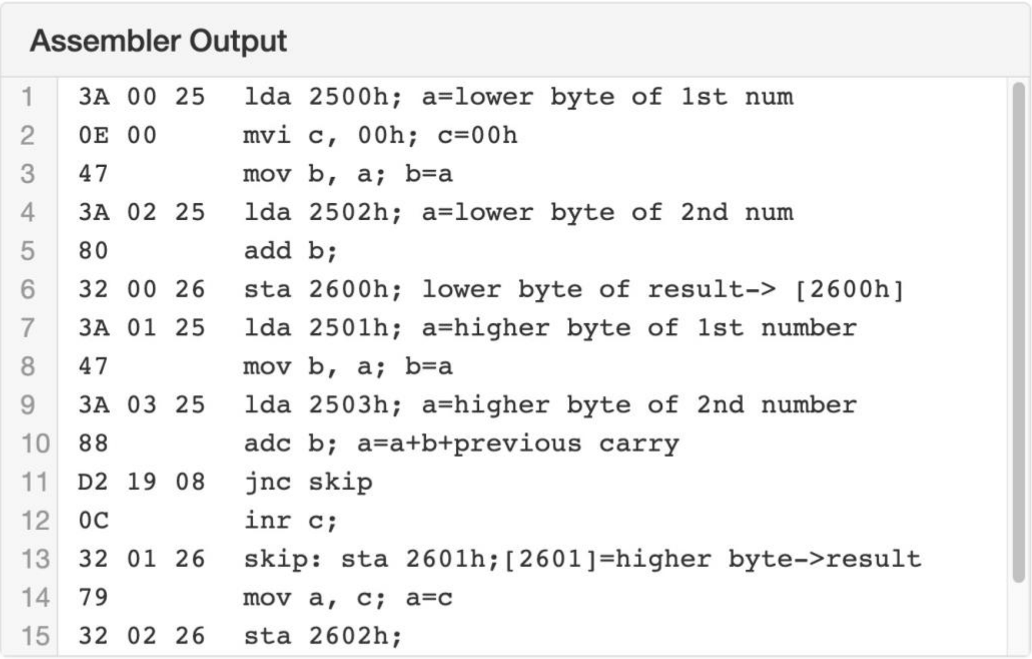
Assembler Output

```
21 00 25     lxi h, 2500h; HL=2500h
7E           mov a, m; a=[2500h]
06 00        mvi b, 00h; stores count of 1
16 08        mvi d, 08h; d=number of bits(8)

0F           back: rrc
D2 0d 08     jnc zero
04           inr b
15           zero:dcr d
C2 08 08     jnz back

78           mov a, b; a=count of 1
32 10 26     sta 2610h; [2610]= count of 1
3E 08        mvi a, 08h; a=08h
90           sub b; a=a-b (count of zeros)
```



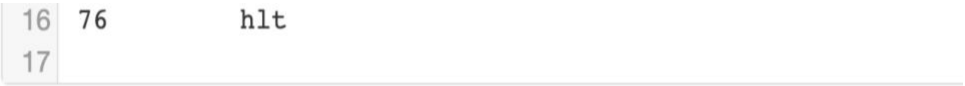
```
32 11 25     sta 2511h;
76           hlt

```

5. Write a program to sum two 16-bits binary numbers.



Assembler Output

```
3A 00 25   lda 2500h; a=lower byte of 1st num
0E 00      mvi c, 00h; c=00h
47         mov b, a; b=a
3A 02 25   lda 2502h; a=lower byte of 2nd num
80         add b;
32 00 26   sta 2600h; lower byte of result-> [2600h]
3A 01 25   lda 2501h; a=higher byte of 1st number
47         mov b, a; b=a
3A 03 25   lda 2503h; a=higher byte of 2nd number
88         adc b; a=a+b+previous carry
D2 19 08   jnc skip
0C         inr c;
32 01 26   skip: sta 2601h;[2601]=higher byte->result
79         mov a, c; a=c
32 02 26   sta 2602h;
```



```
76         hlt
```